```
from google.colab import drive
drive.mount('/content/drive')

     Mounted at /content/drive
```

```
cd /content/drive/MyDrive/CSE475/project1

     /content/drive/MyDrive/CSE475/project1
```

```
import pandas as pd
import matplotlib.pyplot as plt

import numpy as np
from sklearn import linear_model
from sklearn.metrics import mean_squared_error,r2_score
from sklearn.linear_model import Ridge
from sklearn.linear_model import LassoCV
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import ElasticNet
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import accuracy_score
```

```
dataset = pd.read_csv('covid_dataset.csv')
dataset
```

	Day	Lab Test	Confirmed case	Death Case
0	2020-04-04	434	9	2
1	2020-04-05	367	18	1
2	2020-04-06	468	35	3
3	2020-04-07	679	41	5
4	2020-04-08	981	54	3
...	...	...	...	...
621	2021-12-16	25203	257	3
622	2021-12-17	16310	191	2
623	2021-12-18	13991	122	4
624	2021-12-19	19332	211	1
625	2021-12-20	19955	260	2

626 rows × 4 columns

```
x= dataset [['Lab Test','Confirmed case']]
y= dataset[['Death Case']]
x_train , x_test ,y_train,y_test = train_test_split (x,y,test_size = 0.3 , random_state = 42
```

```
lasso_model=linear_model.Lasso()
lasso_model.fit(x_train,y_train)
```

```
     Lasso()
```

```
lasso_model.intercept_
```

```
     array([-4.41198328])
```

```
lasso_model.coef_
```

```
     array([0.00051562, 0.01590487])
```

```
#we determine the alpha according to the different lambda values
from sklearn.linear_model import Lasso
lasso = Lasso()
coefs = []
alphas = np.random.randint(0,1000,100)

# 10**np.linspace(10,-2,100)*0.5 -- It can be used instead of alphas and observed.

for a in alphas:
    lasso.set_params(alpha = a)
    lasso.fit(x_train,y_train)
    coefs.append(lasso.coef_)
```

```
lasso_model.predict(x_train)[:5]
```

```
     array([22.11655481, 64.84567135, 51.98185537, 45.12226283, 27.87136441])
```

```
lasso_model.predict(x_test)[:5]
```

```
     array([ 74.44936591, 275.00439839,  32.7350311 ,  29.84479005,
            265.42401616])
```

```
y_pred = lasso_model.predict(x_test)
np.sqrt(mean_squared_error(y_test,y_pred))
```

```
     17.78687602021199
```

```
r2_score(y_test, y_pred)
```

```
     0.8943961638378569
```

```
lasso_model.score(x_test,y_test)
```

```
     0.8943961638378569
```

Ridge Regression

```
x1= dataset [['Lab Test','Confirmed case']]
y1= dataset[['Death Case']]
x1_train , x1_test ,y1_train,y1_test = train_test_split (x1,y1,test_size = 0.3 , random_state
```

```
ridge_model=linear_model.Ridge()
ridge_model.fit(x1_train,y1_train)
```

```
     Ridge()
```

```
ridge_model.coef_
```

```
     array([[0.00051553, 0.01590523]])
```

```
print("Intercept : %.2f" % ridge_model.intercept_)
```

```
     Intercept : -4.41
```

```
ridge_model.predict(x1_train)[:5]
```

```
     array([[22.11677651],
            [64.84613223],
            [51.98075982],
            [45.12280958],
            [27.87113968]])
```

```
ridge_model.predict(x1_test)[:5]
```

```
     array([[ 74.45219277],
            [275.0101687 ],
            [ 32.73517255],
            [ 29.84597799],
            [265.4304146 ]])
```

```
y1_pred = ridge_model.predict(x1_test)
np.sqrt(mean_squared_error(y1_test,y1_pred))
```

```
     17.78709823171557
```

```
r2_score(y1_test, y1_pred)
```

```
0.8943935252031947
```

```
ridge_model.score(x1_test,y1_test)
```

```
0.8943935252031947
```